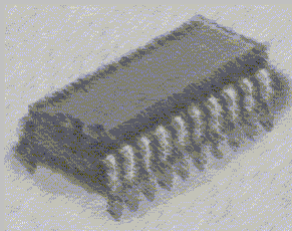


Tema 1. Fundamentos del lenguaje ensamblador



**Laboratorio de Estructura y
Organización de Computadores**

Grados en Ingeniería Informática e
Ingeniería de Computadores

Curso 2012-2013

Tema 1: Fundamentos del lenguaje ensamblador

Índice

- Estructura del 8086
- Tipos de instrucciones
- Instrucciones de transferencia de datos
- Instrucciones aritméticas
- Filosofía de programación
- Estructura de un programa en ensamblador
- Pasos para la creación de un programa ejecutable
- Mi primer programa: edición, ensamblado, enlazado y depuración



Departamento de Automática
Área de Arquitectura y Tecnología de Computadores

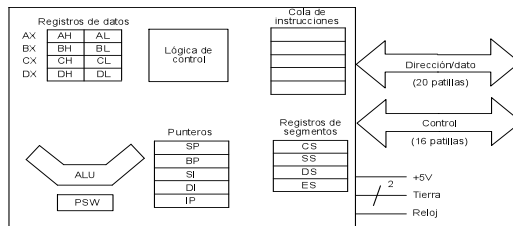
Laboratorio de Estr. y Org. de Computadores
Grados en I. Informática / I. de Computadores

Trans.:
2 / 27

Tema 1: Fundamentos del lenguaje ensamblador

Estructura del i8086 (I)

- El microprocesador 8086 tiene catorce registros de 16 bits:
 - Registros de datos
 - Registros punteros de la pila
 - Registro de instrucciones
 - Registros de segmento
 - Registros índices
 - Registro de flags de estado



Departamento de Automática
Área de Arquitectura y Tecnología de Computadores

Laboratorio de Estr. y Org. de Computadores
Grados en I. Informática / I. de Computadores

Trans.:
3 / 27

Tema 1: Fundamentos del lenguaje ensamblador

Estructura del i8086 (y II)

- Registros de datos:
 - AX (AH, AL)
 - BX (BH, BL)
 - CX (CH, CL)
 - DX (DH, DL)
- Registros punteros:
 - SP - Puntero de pila
 - BP - Puntero base de pila
 - SI - Registro índice
 - DI - Registro índice
 - IP - Contador de programa
- Registro de estado del procesador:
 - PSW
- Registros de segmento:
 - CS - segmento de código
 - DS - segmento de datos
 - ES - segmento extra
 - SS - segmento pila



Departamento de Automática
Área de Arquitectura y Tecnología de Computadores

Laboratorio de Estr. y Org. de Computadores
Grados en I. Informática / I. de Computadores

Trans.:
4 / 27

Tema 1: Fundamentos del lenguaje ensamblador

Tipos de instrucciones del i8086)

Tipo de instrucción	Efecto
Instrucciones de transferencia	Mueven información entre registros, registros y posiciones de memoria, o entre registros y puertos de entrada/salida
Instrucciones aritméticas	Realizan operaciones aritméticas: sumas, restas, etc.
Instrucciones de manejo de bits	Realizan operaciones de desplazamiento, rotación y lógicas sobre registros o posiciones de memoria
Instrucciones de transferencia de control	Sirven para controlar la ejecución de las instrucciones del programa
Instrucciones de entrada salida	Mueven información entre registros y puertos de entrada/salida
Instrucciones de manejo de cadenas	Realizan operaciones sobre cadena de bytes o palabras
Instrucciones de interrupción	Provocan que el microprocesador realice un servicio que se le solicita diferente a las instrucciones que está ejecutando



Tema 1: Fundamentos del lenguaje ensamblador

Instrucciones de transferencia de datos (I)

- **Nombre:** MOV
- **Formato:** MOV destino, origen
- **Descripción:**
Transfiere un byte o una palabra desde el operando origen al operando destino
- **Ejemplos:**
 - MOV CX, 112h ; CX = 112h
 - MOV BL, 255 ; BL = 255 = FFh
 - MOV ES, AX ; ES = AX
 - MOV AL, 12h ; AL = 12h
 - MOV PAL_MEM, BX ; PAL_MEM = BX



Instrucciones de transferencia de datos (II)

- **Nombre:** PUSH
- **Formato:** PUSH origen
- **Descripción:**
Decrementa el puntero de pila (SP) en 2 y luego transfiere la palabra que se ha especificado en el operando origen a lo alto de la pila
- **Ejemplo:**
 - PUSH BX ; Pone BX en la pila



Instrucciones de transferencia de datos (y III)

- **Nombre:** POP
- **Formato:** POP destino
- **Descripción:**
Transfiere un byte o una palabra desde la cima de la pila al operando destino y luego incrementa la pila en 2
- **Ejemplo:**
 - POP BX ; Pone en BX el contenido de la cima de la pila



Tema 1: Fundamentos del lenguaje ensamblador

Instrucciones aritméticas (I)

- **Nombre:** ADD
- **Formato:** ADD destino, origen
- **Descripción:**
Suma los dos operandos y el resultado lo deja en el operando destino. Los operandos deben ser del mismo tipo
- **Ejemplos:**
 - ADD CL, BL ; CL = CL + BL
 - ADD AL, 12h ; AL = AL + 12h
 - ADD CX, DX ; CX = CX + DX



Departamento de Automática
Área de Arquitectura y Tecnología de Computadores

Laboratorio de Estr. y Org. de Computadores
Grados en I. Informática / I. de Computadores

Trans.:
9 / 27

Tema 1: Fundamentos del lenguaje ensamblador

Instrucciones aritméticas (II)

- **Nombre:** ADC
- **Formato:** ADC destino, origen
- **Descripción:**
Suma los dos operandos más el posible acarreo de la operación anterior. El resultado se almacena en el operando destino. Además los operandos deben ser del mismo tipo
- **Ejemplos:**
 - ADC CL, BL ; CL = CL + BL + CF
 - ADC AL, 12h ; AL = AL + 12h + CF
 - ADC CX, DX ; CX = CX + DX + CF



Departamento de Automática
Área de Arquitectura y Tecnología de Computadores

Laboratorio de Estr. y Org. de Computadores
Grados en I. Informática / I. de Computadores

Trans.:
10 / 27

Tema 1: Fundamentos del lenguaje ensamblador

Instrucciones aritméticas (III)

- **Nombre:** SUB
- **Formato:** SUB destino, origen
- **Descripción:**
Resta el operando origen del operando destino. El resultado se almacena en el operando destino y además, ambos operandos deben ser del mismo tipo
- **Ejemplos:**
 - SUB CL, BL ; CL = CL - BL
 - SUB AL, 12h ; AL = AL - 12h
 - SUB CX, DX ; CX = CX - DX



Departamento de Automática
Área de Arquitectura y Tecnología de Computadores

Laboratorio de Estr. y Org. de Computadores
Grados en I. Informática / I. de Computadores

Trans.:
11 / 27

Tema 1: Fundamentos del lenguaje ensamblador

Instrucciones aritméticas (IV)

- **Nombre:** SBB
- **Formato:** SBB destino, origen
- **Descripción:**
Resta el operando origen del operando destino. Resta uno si el flag de acarreo está activo. Los operandos deben ser del mismo tipo. El resultado se almacena en el operando destino
- **Ejemplo:**
 - SBB CX, DX ; CX = CX - DX - CF



Departamento de Automática
Área de Arquitectura y Tecnología de Computadores

Laboratorio de Estr. y Org. de Computadores
Grados en I. Informática / I. de Computadores

Trans.:
12 / 27

Tema 1: Fundamentos del lenguaje ensamblador

Instrucciones aritméticas (V)

- **Nombre:** MUL
- **Formato:** MUL origen
- **Descripción:**
Multiplica, sin considerar el signo el acumulador (AL o AX) por el operando origen.
Si el operando origen es de tipo byte, lo multiplica por AL y el resultado se almacena en AX. Si es de tipo palabra, lo multiplica por AX y el resultado se almacena en DX (palabra superior) y AX (palabra inferior)
- **Ejemplo:**
 - ; AX = 1234h
 - ; BX = 1000h
 - MUL BX ; DX = 0123h, AX = 4000h



Departamento de Automática
Área de Arquitectura y Tecnología de Computadores

Laboratorio de Estr. y Org. de Computadores
Grados en I. Informática / I. de Computadores

Trans.:
13 / 27

Tema 1: Fundamentos del lenguaje ensamblador

Instrucciones aritméticas (VI)

- **Nombre:** IMUL
- **Formato:** IMUL origen
- **Descripción:**
Multiplica, considerando el signo, el acumulador AL o AX por el operando origen.
Si el operando fuente es un byte, lo multiplica por AL y se almacena el resultado en AX. Si se trata de una palabra, lo multiplica por AX y el resultado se almacena en DX (palabra superior) AX (palabra inferior)
- **Ejemplos:**
 - ; AL = FEh = -2
 - ; BL = 12h = 18
 - IMUL BL ; AX = FFDCh = -36



Departamento de Automática
Área de Arquitectura y Tecnología de Computadores

Laboratorio de Estr. y Org. de Computadores
Grados en I. Informática / I. de Computadores

Trans.:
14 / 27

Tema 1: Fundamentos del lenguaje ensamblador

Instrucciones aritméticas (VII)

- **Nombre:** DIV
- **Formato:** DIV origen
- **Descripción:**
Divide, sin considerar el signo, el acumulador AL o AX y su extensión (AH o DX) por el operando origen. Si el operando origen es un byte, divide AX por dicho operando y el resultado se almacena en AL y el resto en AH. Si el origen es una palabra, divide DX y AX entre dicho operando y el resultado se almacena en AX y el resto en DX.
- **Ejemplos:**
 - ; AX = 0013h = 19
 - ; BL = 02h = 2
 - DIV BL ; AH = 1, AL = 9



Departamento de Automática
Área de Arquitectura y Tecnología de Computadores

Laboratorio de Estr. y Org. de Computadores
Grados en I. Informática / I. de Computadores

Trans.:
15 / 27

Tema 1: Fundamentos del lenguaje ensamblador

Instrucciones aritméticas (VIII)

- **Nombre:** IDIV
- **Formato:** IDIV origen
- **Descripción:**
Divide, considerando el signo, el acumulador AL o AX y su extensión (AH o DX) por el operando origen. Si el operando origen es un byte, divide AX por dicho operando y el resultado se almacena en AL y el resto en AH. Si el origen es una palabra, divide DX y AX entre dicho operando y el resultado se almacena en AX y el resto en DX.
- **Ejemplos:**
 - ; AX = FFEDh = -19
 - ; BL = 02h = 2
 - IDIV BL ; AH = 1, AL = F7h = -9



Departamento de Automática
Área de Arquitectura y Tecnología de Computadores

Laboratorio de Estr. y Org. de Computadores
Grados en I. Informática / I. de Computadores

Trans.:
16 / 27

Tema 1: Fundamentos del lenguaje ensamblador

Instrucciones aritméticas (IX)

- **Nombre:** INC
- **Formato:** INC destino
- **Descripción:**
Suma una unidad al operando destino. El operando puede ser de tipo byte o palabra
- **Ejemplos:**
 - ; AX = 1234h
 - INC AX ; AX = 1235h
 - INC AH ; AH = 13h



Departamento de Automática
Área de Arquitectura y Tecnología de Computadores

Laboratorio de Estr. y Org. de Computadores
Grados en I. Informática / I. de Computadores

Trans.:
17 / 27

Tema 1: Fundamentos del lenguaje ensamblador

Instrucciones aritméticas (X)

- **Nombre:** DEC
- **Formato:** DEC destino
- **Descripción:**
Resta una unidad al operando destino. El operando puede ser de tipo byte o palabra
- **Ejemplos:**
 - ; AX = 1234h
 - DEC AX ; AX = 1233h
 - DEC AH ; AH = 11h



Departamento de Automática
Área de Arquitectura y Tecnología de Computadores

Laboratorio de Estr. y Org. de Computadores
Grados en I. Informática / I. de Computadores

Trans.:
18 / 27

Tema 1: Fundamentos del lenguaje ensamblador

Instrucciones aritméticas (y XI)

- **Nombre:** NEG
- **Formato:** NEG destino
- **Descripción:**
 - Cambia de signo mediante el complemento a 2 del operando destino. Deja el resultado en el operando destino. El operando puede ser de tipo byte o palabra
- **Ejemplo:**
 - NEG AL ; Si AL = F2h antes de la operación, después AL = 0Eh



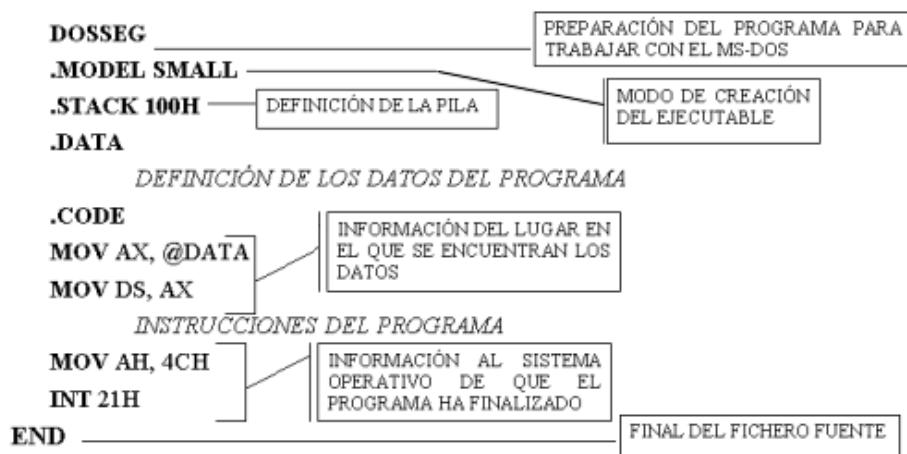
Departamento de Automática
Área de Arquitectura y Tecnología de Computadores

Laboratorio de Estr. y Org. de Computadores
Grados en I. Informática / I. de Computadores

Trans.:
19 / 27

Tema 1: Fundamentos del lenguaje ensamblador

Estructura de un programa en ensamblador



Departamento de Automática
Área de Arquitectura y Tecnología de Computadores

Laboratorio de Estr. y Org. de Computadores
Grados en I. Informática / I. de Computadores

Trans.:
20 / 27

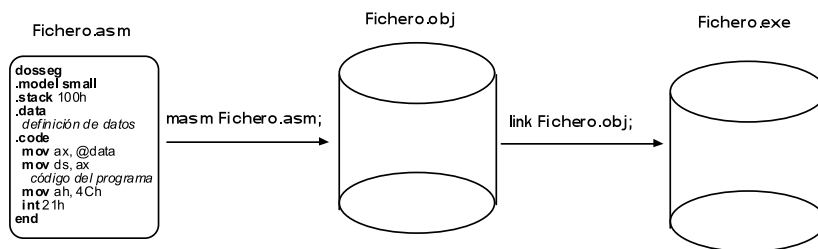
Tema 1: Fundamentos del lenguaje ensamblador

Pasos para crear un programa ejecutable

Programación:

- Los programas deben escribirse en un editor de texto ASCII
- El nombre del fichero debe tener como extensión ASM
- Para ensamblar el fichero se debe teclear en la línea de órdenes:
MASM NombreFichero.ASM
- Si no se producen errores, se debe enlazar tecleando en la línea de órdenes:

LINK NombreFichero.OBJ



Departamento de Automática
Área de Arquitectura y Tecnología de Computadores

Laboratorio de Estr. y Org. de Computadores
Grados en I. Informática / I. de Computadores

Trans.:
21 / 27

Tema 1: Fundamentos del lenguaje ensamblador

Mi primer programa en ensamblador (I) Edición del programa primero.asm

```
dosseg
.model small
.stack 100h
.data
    num1      db 12h
    num2      db 10h
.code
    mov ax, @data
    mov ds, ax
    mov al, num1
    mov bl, num2
    mul bl
    mov ah, 4Ch
    int 21h
end
```



Departamento de Automática
Área de Arquitectura y Tecnología de Computadores

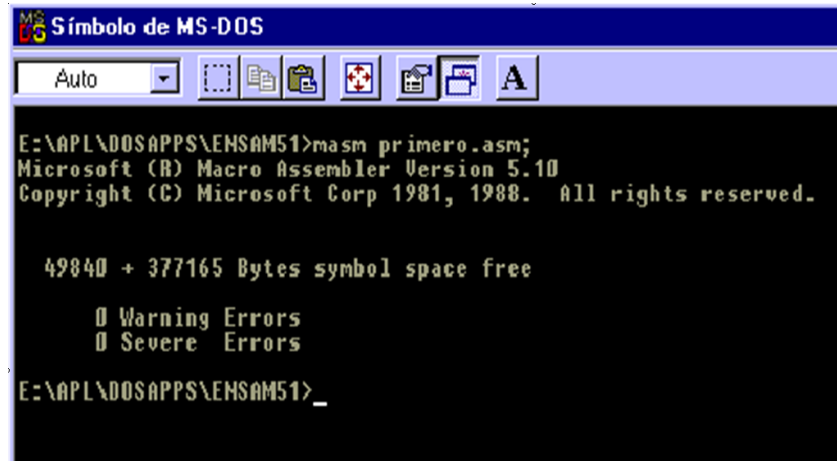
Laboratorio de Estr. y Org. de Computadores
Grados en I. Informática / I. de Computadores

Trans.:
22 / 27

Tema 1: Fundamentos del lenguaje ensamblador

Mi primer programa en ensamblador (II)

Ensamblado del programa primero.asm



```
MS-DOS Símbolo de MS-DOS
Auto
E:\APL\DOSAPPS\ENSAM51>masm primero.asm;
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

49840 + 377165 Bytes symbol space free

0 Warning Errors
0 Severe Errors

E:\APL\DOSAPPS\ENSAM51>_
```



Departamento de Automática
Área de Arquitectura y Tecnología de Computadores

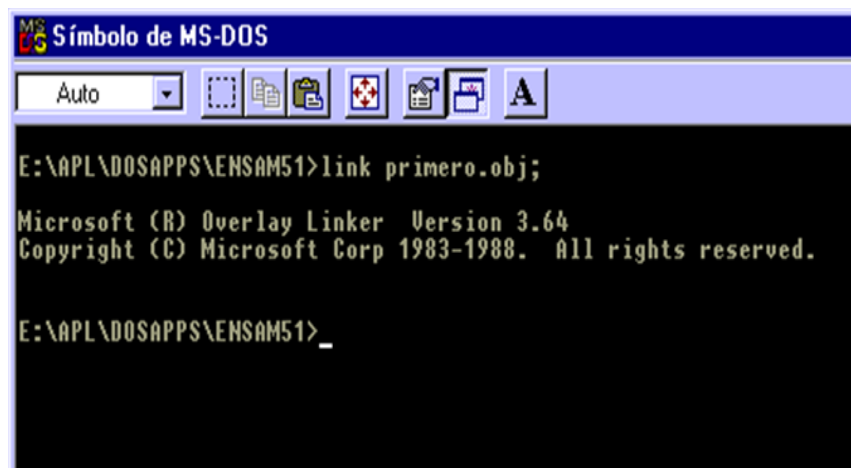
Laboratorio de Estr. y Org. de Computadores
Grados en I. Informática / I. de Computadores

Trans.:
23 / 27

Tema 1: Fundamentos del lenguaje ensamblador

Mi primer programa en ensamblador (III)

Enlazado del programa primero.obj



```
MS-DOS Símbolo de MS-DOS
Auto
E:\APL\DOSAPPS\ENSAM51>link primero.obj;
Microsoft (R) Overlay Linker Version 3.64
Copyright (C) Microsoft Corp 1983-1988. All rights reserved.

E:\APL\DOSAPPS\ENSAM51>_
```



Departamento de Automática
Área de Arquitectura y Tecnología de Computadores

Laboratorio de Estr. y Org. de Computadores
Grados en I. Informática / I. de Computadores

Trans.:
24 / 27

Tema 1: Fundamentos del lenguaje ensamblador

Mi primer programa en ensamblador (IV)

Ejecución paso a paso con el Code View (I)

Depuración:

- Si los programas no funcionan correctamente o se desea observar su funcionamiento se pueden ejecutar paso a paso
- El programa que permite realizar esta operación es el CODEVIEW
- Sintaxis CV NombreFichero.EXE
- Con F2 se activa o desactiva la ventana de registros
- Con F8 se ejecuta paso a paso y comprueba las subrutinas también
- Con F10 se ejecuta paso a paso y se salta la comprobación de las subrutinas



Departamento de Automática
Área de Arquitectura y Tecnología de Computadores

Laboratorio de Estr. y Org. de Computadores
Grados en I. Informática / I. de Computadores

Trans.:
25 / 27

Tema 1: Fundamentos del lenguaje ensamblador

Mi primer programa en ensamblador (y V)

Ejecución paso a paso con el Code View (y II)

The screenshot shows the MS-DOS CodeView window. The main window is divided into three panes. The left pane shows the assembly code with line numbers 1 through 18. The middle pane shows the current instruction being executed, which is highlighted in blue. The right pane shows the state of the 8086 registers. The registers are listed on the right side of the window, with their current values. The registers are: AX = 0000, BX = 0000, CX = 0000, DX = 0000, SP = 0100, BP = 0000, SI = 0000, DI = 0000, DS = 4E28, ES = 4E28, SS = 4E38, CS = 4E38, IP = 0010. The status bar at the bottom of the window shows 'Microsoft (R) CodeView (R) Version 2.2 (C) Copyright Microsoft Corp. 1986-1988. All rights reserved.'

Zona de código

Registros del i8086

Zona de órdenes



Departamento de Automática
Área de Arquitectura y Tecnología de Computadores

Laboratorio de Estr. y Org. de Computadores
Grados en I. Informática / I. de Computadores

Trans.:
26 / 27

Tema 1: Fundamentos del lenguaje ensamblador

Enlace a CodeVIEW y resumen instrucciones

- http://atc2.aut.uah.es/personal_archivos/ACastejon.html
 - **Guía de referencia rápida del CodeView.**
 - **Programa MASM5.1**
 - **Guía de referencia rápida del repertorio del 8086.**



Departamento de Automática
Área de Arquitectura y Tecnología de Computadores

Laboratorio de Estr. y Org. de Computadores
Grados en I. Informática / I. de Computadores

Trans.:
27 / 27